#5

2131

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

JOHAN C. TALSTRA ET AL

Atty. Docket No.

NL000262

Serial No. 09/853,174

Group Art Unit: 2131

Filed: MAY 10, 2001

Title: COPY PROTECTION SYSTEM

Commissioner for Patent
Washington, D.C. 20231

CLAIM FOR PRIORITY

Sir:

A certified copy of the European Application No. 00201669.9 filed May 10, 2000 and referred to in the Declaration of the above-identified application is attached herewith.

Applicants claim the benefit of the filing date of said European application.

Respectfully submitted,

By Michael E Belk

Enclosure

Michael E. Belk, Reg. 33,357
Attorney
(914) 333-9643

CERTIFICATE OF MAILING

It is hereby certified that this correspondence is being deposited with the United States Postal Service as first-class mail in an envelope addressed to:
COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

On August 3, 2001

By Neemi Chopra

S:\be\MH25BEB0.NC0.doc

NL000262
US

Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

00201669.9

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 07/05/01
LA HAYE, LE

Europäisches Patentamt | European Patent Office | Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 00201669.9
Demande n°:

Anmeldetag:
Date of filing: 10/05/00
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.
5621 BA Eindhoven
NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Physical disc mark trigger in encrypted content

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

Physical disc mark trigger in encrypted content

Introduction

Films released on DVD are protected from being copied by the so-called CSS encryption method, well known to a person skilled in the art. In the future, additional protection methods such as digital watermarking will be added. With the imminent introduction of recordable and rewritable DVD formats into the consumer-market, there is also the need of so called "play control" which ensures that certain copy protection rules are checked. One of these rules is the following: CSS encrypted content on a recordable disc should be refused. This rule has been specified in the CSS-license, signed by all DVD-manufacturers, but has not been substantiated in its technical realisation. In other words, although all DVD-player manufacturers should obey this rule per the CSS-license, there is no clear way to implement this. The invention disclosed here presents such a realisation.

In order to implement this rule, recordable discs have to be distinguished from pre-recorded discs, e.g. DVD-ROM discs. There are two ways of approaching this problem:

- Recognise all recordable formats (present and future) (e.g. pre-groove detection). This method is technically simple but seriously flawed from a security point of view. There is an incentive for recordable disc manufacturers to continually attempting to modify their recordable media in such a way that players (not recorders) recognise them as ROM discs, so as to legally circumvent the CSS-rule. New players would have to recognise those new discs as well, i.e. an arms race.
- Introduce a physical disc mark for DVD-ROM discs which cannot be reproduced by consumers on recordable discs e.g. ROM-wobble. This wobble is a (small) radial variation of the spiral made up by pits and lands and recorded in phase. This wobble can be detected in a player from the DPD-radial servo-tracking signal, present in the basis engine. The discs upon which such a wobble is detected are marked pre-recorded, whereas discs without a wobble are marked recordable. In this way, the wobble can be used for distinguishing pre-recorded discs from recordable discs.

In the second solution, for additional security, the proposed ROM-wobble can have a payload, which is (cryptographically) tied to the content, e.g. by using the payload in the watermark. This is where the wobble shows its real strength. The wobble could also be tied to CSS, which has the added bonus of providing an upgrade path.

The problem with introducing the ROM-wobble is the presence of legacy ROM-discs with CSS content that do not have the wobble. I.e. there are 2 types of discs without a wobble: i) recordable or rewriteable discs which should be rejected when comprising protected content, e.g. CSS protected content, ii) legacy pre-recorded discs which should be played back (even when comprising (CSS) protected content). Therefore, it is required that in the content on "new" discs there will be a "wobble-trigger" (as well as the payload). This trigger has the following requirements:

- It should be easily detectable from looking just at the content
- It should not be easily removable by a hacker
- It should not affect content preparation

Results known thus far; limitations

Previous solutions did not meet all of the above criteria. Watermarks embedded in the video are not easily detectable: the content is CSS-encrypted, and checking for the watermark requires decryption, which is typically expensive in a DVD-drive.

An alternative watermark method on the level of the MPEG stream (so called PTY marks) is easily detected, but is not acceptable from the viewpoint that the impact on content preparation should be low.

Straightforward methods of setting a few bits in the CSS encrypted content are easily hacked.

Proposed solution i.

CSS-encrypted content is typically decrypted both in hardware (in tabletop DVD-players) and software (in PC's). Software decryption slows down the PC substantially, and seriously degrades the viewing quality of the DVD-film. To ameliorate this situation, only a limited fraction of the video stream has been encrypted in the DVD-mastering facility. The stream is divided into so called packs of 2 Kbytes each, and typically somewhere between 10-50% of the packs have been encrypted. The invention is based on the recognition that a message for the purpose of copy protection may be transmitted by the deliberately

encrypting packs following a certain pattern. As an example, encrypt the packs according to the rule:

u – u – u – e – e – u – u – u – e – e – u – u – u – e – e – u – u – u – e – e – ....

to transmit a '0' message, and

u – u – u – u – e – e – u – u – u – u – e – e – u – u – u – u – e – e – ....

to transmit a '1' bit, where 'u' stands for an unencrypted pack, and 'e' for an encrypted one. For a hacker to remove these messages (which would be interpreted by the DVD-player in accordance with the purpose of this invention to expect an appropriate disc-mark like the wobble) he would need to decrypt CSS and re-encrypt it; decryption is not enough, because the watermark can be detected in clear content. The particular manner to encode information in the pattern of encrypted/unencrypted packs should be sufficiently exotic that it has an extremely low probability of having occurred in DVD encoded in the past. Therefore something like pseudo-random noise patterns of u's and e's would be more suitable.

Biased pseudo random noise sequences

Because the number of encrypted and unencrypted packs per second is not equal (the number of 'u''s is usually quite larger than 'e''s to facilitate DVD-playback in software) the aforementioned pseudo-random patterns would have to be biased somehow. The standard manner to cheaply construct a pseudo-random noise sequence is the LFSR (linear feedback shift register), which is defined by a so-called irreducible generator polynomial of a finite field $GF(p^q)$, where q is the length of the LFSR. It is common to choose p = 2. However to create a biased pseudo-random sequence with bias 1/s (i.e. out of every s packs, s-1 are unencrypted and 1 is encrypted), with s prime, one should choose the polynomial over GF(s). The output of the LFSR is then a random sequence of elements l_i of GF(s): 0, 1, 2, ..., s-1. If we replace every l_i by 'u' if $l_i \geq 1$, and by 'e' if $l_i = 0$, otherwise, we obtain a recipe to encrypt the packs with the required bias.

Proposed solution ii.

The keys used to encrypt the content are 40 bits long. The second solution consists of designing a function operating on the key $K \rightarrow f(K)$, where f(K) can be 0 or 1. f( ) has to be chosen in such a way that when operating on the keys used in the DVD-titles published so far (on the order of 4000 keys), it always yields 0. The way to enforce the CSS-rule would then be that a player reads the disc key K, computes f(K), and if the result is 0, it knows that no wobble is necessary (because the key must belong to a movie published in a

time when the wobble was not required yet). If the result however is '1', then the player must also check for a wobble. If there is no wobble, the disc is an illegal copy of CSS-encrypted material on a recordable, or illegitimately mastered ROM disc.

After introduction of this system, the implication for the publishers is that before encrypting a movie with key K, they would check whether f(K)=1 when they want wobble protection for their content, and f(K)=0 when they don't. If the key K doesn't have the appropriate properties, a new random K needs to be chosen. In practice this is not a problem, because disc-keys are distributed by a single licensing organisation the "DVD_CCA", located in California.

For this reason a desirable property of f( ) would be that it would be 0 on one half of all possible keys and 1 on the other half; in that case on average no more than 2 tries are needed to find a suitable K. There is an additional reason to require f( ) to have this property: f( ) would be built into DVD-players and would therefore potentially be known publicly. It would be undesirable if the keys of all past 4000 DVD titles could be derived from knowing f( ) alone. In the section below we will explain how such a function can be constructed from a given set of 4000 arbitrary keys. The conclusion is that f( ) is surprisingly simple a) to compute and b) to implement. Implementation requires storage of approximately 64 40-bit (non-confidential) constants, and computation requires 7 40-bit XOR operations + 1 shift register.

Efficient derivation of f( )

The derivation of the function f is based on a mathematical result that can be stated roughly as follows. Further mathematical details can be found in Appendix A.

If X is a collection of m-bit keys, of size n, say, then there exists an m-bit number a such that if we partition the collection X into two parts according to the value of the XOR of elements from X with a, then each of the parts contains about half of the elements of X. If a is chosen at random, then for each e>1, the probability that the sizes of both parts differ from n/2 by at most e.sqrt(n) is at least 1/(1-e). Also, if n<m, then there is an a such that the XOR of a with all elements from X is 0.

Using this result, we can construct a function f such that the evaluation of f(K) can be arranged in the form of a binary decision-tree of depth d with d approximately equal to log(n) – log(m), where log() denotes the base-2 logarithm. Here, in each node v of the decision-tree, we compute the m-bit XOR of K with the m-bit number a(v) corresponding to this node; we let the result of this XOR determine which of the two branches from v will be

followed. The value of f(K) will be the computed XOR-value at the end-node that is reached after d steps.

In the above practical case, we have n = 4000 and m=40, so that d is about 7. The decision- tree will contain 2^d-1, so about 127, nodes, which means that we will have to store about 127 40-bit numbers a while an evaluation of f will require about d = 7 m-bit XOR's.

Both proposed solutions have as an advantage that the "wobble trigger" does not need decryption and watermark detection. This is accomplished by embedding the trigger, used to distinguish new, wobbled media from legacy discs, in the encryption instead of in the watermark.

The solutions have as additional advantages:

- Wobbled discs play on legacy players;
- The encrypted content on wobbled discs contains a secure, wobble trigger which is hard to remove;
- Legacy discs play on new players, because the wobble trigger is not present, so the player will not check on the existence of a wobble. As a result the wobbled discs and the not-wobbled discs can co-exist;
- The wobble provided an optional extra level of security;
- The wobble works with CPPM (Copy Protection for Pre-recorded Media; the copy protection scheme for DVD-Audio) or CSS;
- Wobble detection in the drive requires limited hardware cost (5-6 KGates).

Although the design of the 2 schemes outlined above has been specifically triggered by problems in the DVD arena, it is conceivable that in particular the second proposed solution in this disclosure has a much wider range of applications. E.g. a revocation scheme could be based on this. A player would have the general structure of the function f( ) on board, but it would load the constants dynamically.

In Figure 1 a schematically drawing of an apparatus for reading out an information carrier is shown.

Figure 1 shows an apparatus according to the invention for reading of the information carrier 17. The apparatus comprises driving means 26 for rotating the information carrier 17 and a read head 27 for reading out the tracks present on the

information carrier. The read head 27 comprises an optical system of a known type to focus a light spot 28 on a track by means of a beam of light 29 guided through optical elements like a collimator lens 39, to collimate the beam of light and an objective lens, to focus the beam of light. This beam of light 29 originates from a radiation source 41, e.g. an infrared laser diode with a wavelength of 650 nm and an optical output of 1 mW. The read head 27 further comprises a tracking actuator for fine-positioning the light spot 28 in the radial direction in the middle of the track. Adjusting the position of the light spot to the position of the track can also be achieved be changing the position of the objective lens 40.

After being reflected by the information carrier 17, the beam of light 29 is detected by a detector 42 of a known type, e.g. a quadrant detector en generates detector signals 31 including a read signal, a tracking-error signal, focussing-error signal, synchronisation signal and lock-in signal. E.g. a beam splitting cube 43, a polarising beam splitting cube, a pellicle or a retarder can be used for this. The apparatus further comprises tracking means 32 connected to the read head 27 for receiving the tracking-error signal of the read head 27 and for steering the tracking actuator 30. During reading out the information carrier 17 the reading-out signal is converted in the read out means 34 into output information 33 the read out means for example comprising a channel decoder or an error-corrector. The apparatus further comprises an address detector 35 for retrieving the addresses from the detector signals 31 and positioning means 36 for coarse positioning the read head 27 in de radial direction of the track. The apparatus further comprises detection means 48 for receiving the detector signals 31 from the read head 27. The detector signals 31 are used by the detection means 48 for synchronising the read out means 34. The apparatus further comprises a system control unit 37 for receiving commands of a controlling computer system or a user and for regulating the apparatus by means of control lines 38, e.g. a system bus connected to the driving means 26, the positioning means 36, the address detector 35, the tracking means 32 and the read out means 34.

In this apparatus for reading out information on an information carrier a check is performed which results in a possible refusal to play back the information carrier if a predefined condition, substantially as described above, is not matched.

With reference to Figure 2 and Figure 3, the following checks can occur in the play back apparatus according to the invention (it must be noted that a legacy disc is a pre-recorded disc comprising encrypted content, a wobbled disc is a pre-recorded disc comprising a wobble, a legacy drive is an old compliant drive, a new drive is a new compliant drive):

- legacy drive + legacy disc → pass;
- legacy drive + wobbled disc → pass (the "old" legacy drive doesn't see the disc-mark, i.e. the wobble, but doesn't notice the wobble trigger either);
- new drive + legacy disc → pass (the new drive doesn't find the disc-mark on the old disc, but no wobble trigger either);
- new drive + wobbled disc → pass (the new drive finds the wobble trigger and also finds the wobble; as an option, to further strengthen the copy protection scheme, the payload of the wobble can be detected and checked);
- new drive + non-legacy disc → fail (the new drive finds the wobble trigger, but doesn't find the wobble, necessary for playing the content on the disc).

It must be noted that the invention as described above is not limited to the embodiments explained. For example, the invention is not only related to DVD ROM-discs, but to all pre-recorded media in general. Further, the invention is not only related to a wobble, but to all physical disc marks, which can be used for distinguishing pre-recorded discs from recordable discs. Further, the invention is not only related to CSS, but to all encryption schemes. Further, the invention is not only related to the triggers as described above, but related to all triggers obeying the following conditions: i) detection of the trigger is possible without decrypting the content, ii) the trigger can not be removed without decrypting the content.

APPENDIX A

Chapter 1

Introduction

1.1 Origin of the problem

In this section, we will explain the origin of the problem that we investigate in this report.

At this time, a certain encryption method is used to encrypt so-called DVD-discs. DVD-discs are CD-Rom-like discs containing for example entire movies and are played by the end user on a DVD-player. We call the data written on the DVD-disc the *content* and the owner of the copy rights of the content the *content provider*.

In the near future, the end user is able to copy prerecorded DVD-discs onto its own recordable DVD-discs. Because this could be a problem for the content providers, there are regulations about the ability of viewing the content of recordable DVD-discs. These regulations say, for example, that a DVD-player should not be able to play certain content when recorded by an end user, that is, when recorded on a recordable DVD-disc.

To comply with the regulations, the DVD-player should have a method to detect whether the disc inserted is a prerecorded disc or a recordable disc. We have developed a method for doing this by inserting a physical disc mark, e.g. a so-called ROM-wobble. This disc mark imposes writing some extra information on the prerecorded DVD-disc, which can not be written by the DVD-writers of the home users. The new DVD-player then can easily detect the nature of the inserted disc by reading that extra information.

Now there is one "little" problem: all the existing prerecorded discs don't

have this extra information, so the new DVD-player would consider any existing disc as a recordable disc and therefore not play it. Now we should design the new DVD-player in such a way that it does not perform the prerecorded/recordable detection if the content is"old", where we call the content that has already been released on prerecorded DVD-discs "old" and the content that we want to write on DVD-discs with the new physical disc mark "new".

In practice, each content has a unique *title key*. This is a 40-bits key attached to the content which can be read by the DVD-player. The DVD-player could use this key to determine whether it is "old" or "new", where the "old" keys are the keys of the "old" content. However, these title keys are not public; therefore the DVD-player may not use the method of keeping a list of the "old" keys and checking the occurence of the key of the inserted disc in this list. Furthermore, the content provider should be able to create more "old" keys, for example for producing DVD-discs of which the copies may be read at all times. In contrast to the "old" keys, the "new" keys may still be chosen, so the DVD-player has to be able to decide whether the proposed key is *surely* "new" or *possibly* "old". After finding an implementation for this, we can find a construction for the "new" keys such that they will be recognized by the DVD-player in the right way.

At the moment the number of title keys of released DVD-discs is about 2000. So this is approximately the amount of "old" keys we have to deal with.

Now we can conclude that we have to implement an algorithm in the DVD-player that performs this "key detection" having the following properties:

1. It has as input one 40-bits key and it has two possible outputs: one saying 'this key is possibly "old" ', the other saying 'this key is surely "new" ';
2. it contains in itself as little information as possible about the "old" keys; and
3. it can be easily implemented in hardware.

Having these precise specifications for the *key detection algorithm* we can look at the problem of finding it in a more sophisticated way.

Let us first consider the set of all possible title keys. In the above case this is equal to the space consisting of all binary vectors of length 40. Let this

space be denoted by V, where $V = \mathbb{F}_2^m$, that is, the m-dimensional vector space over $\mathbb{F}_2$, where $m = 40$. Throughout this report $\mathbb{F}_q$ will denote the Galois Field, or, equivalently, the finite field of order q. Note that q has to be a prime or one of its powers.

Now let the given set of "old" keys be denoted by X, that is, we have $X \subseteq V$ and let n be its size, that is, $n = |X|$. As above, in practice we have $n \approx 2000$.

The first two desired properties of the key detection algorithm leads us to the following idea: perhaps we can easily partition the space of possible keys in two equal sized halves, one of which containing all the "old" keys, the other none of them. The key detection algorithm then only has to detect whether the input key is in the "old half" or not. Besides, this makes it very easy to find new keys, even new "old" ones; one can just pick vectors randomly until a right one is found.

This idea together with the third desired property could give us the following idea for a solution: try to find the *hyperplane* H of V that contains all the vectors from X and use it as the "old half". To understand this idea, let us first explain what an hyperplane is and give some of its properties.

A hyperplane of a general vector space V is a linear subspace of V that has dimension one less than the dimension of V. Each hyperplane is uniquely defined by the vector in V orthogonal to it and, conversively, each vector a in $V \setminus \{0\}$ uniquely defines the hyperplane $\{v \in V | (a, v) = 0\}$. Note that $(\cdot, \cdot)$ denotes the inner product in the vector space V. Note also that, in contrast to many infinite fields, in vector spaces over finite fields there exist vectors that are orthogonal to itself. Take for example the vector $[1, 1]^\top \in \mathbb{F}_2^2$; we have that $([1, 1]^\top, [1, 1]^\top) = 1 \cdot 1 + 1 \cdot 1 = 1 + 1 = 0$. (We consider vectors as being *column vectors*.) Another property of a hyperplane in a vector space V over a finite field $\mathbb{F}_q$ is that it consists of the q-th part of the vectors from V. So any hyperplane of $V = F_2^m$ consists of exactly half of the elements of V.

Let us now return to the idea as stated above. We can see now that this idea about using a hyperplane for the partition of V has two of the three desired properties: firstly, it partitions V in exactly two equal sized halves, and secondly, testing whether the input key is in the hyperplane or not is done by just calculating the inner product of the input key with the vector defining the hyperplane. The latter property makes this idea extremely easy to implement in hardware.

However, can we surely find a hyperplane that contains all the vectors

from X, where $X \subseteq V = \mathbb{F}_2^m$? The answer is no, we cannot. It is only possible when X lies in an $(m-1)$-dimensional subspace of V, which is the case if the span of X has an dimension of $(m-1)$ or less. Only if $|X| \leq m-1$ we are sure this is the case, and if $|X| \gg m$ then it is very likely that the span of X has dimension equal to m, or, equivalently, that X spans V.

So the above idea does only work for sure if $|X| < m$. Now we can think of the following solution: divide V in several subsets $V_1, \ldots, V_T$ such that each subset contains a sufficiently small number of the elements from X, that is, $|X \cap V_i| < m$ for all i. Then apply the above idea on those subsets, that is, find for each subset V_i a hyperplane H_i such that $X \cap V_i \subset H_i$. Now the key detection algorithm consists of two steps: firstly it determines in which subset V_i of V the input key lies and secondly it computes the inner product of the input key with the vector defining the hyperplane H_i.

The greatest problem now is how to divide V into T subsets in a smart way. Of course, we can again use hyperplanes to perform this partition, namely by the following *partition steps*. In the first step ($t = 1$) we divide the m-dimensional vector space V by a hyperplane H into $2^t = 2$ parts $V_0 = H$ and $V_1 = V \backslash H$. After that we divide the obtained 2^t subsets by (different) hyperplanes, thus creating a new partition of V into 2^{t+1} subsets. We can repeat this step several times until, after t steps in total, the obtained 2^t subsets are small enough. Note that we can consider the obtained subsets after t steps as $(m-t)$-dimensional subspaces of V (we need to translate the origin of the vector space V for some subsets to have this property). Note also that each subset is uniquely defined by the $1+2+4+\cdots+2^{t-1} = 2^t - 1$ subsequent hyperplanes which are used to partition V. These hyperplanes are again uniquely defined by $2^t - 1$ vectors from V.

It is clear that the total partition would be optimal if the subsets each contain approximately the same number of elements from X. This can be done by choosing the partitioning hyperplanes in such way that they also partition the corresponding subset of X into two equal halves. For example, in the first step we have to make sure that $|V_0 \cap X| \approx |V_1 \cap X|$, or equivalently, that $|X \cap H|$ is close to $|X|/2$.

Take for example $X \subseteq V = \mathbb{F}_2^{40}$ with $|X| = 2000$, values that in practice are expected. After 6 times partitioning V, we have constructed $2^6 = 64$ subsets of V which have in the ideal case all an intersection with X of size approximately $2000/2^6 \approx 32 < 40$. In this case the key detection algorithm has built in the 63 vectors defining the hyperplane partitioning of V into 64 subsets, which it can use to determine in which subset V_i the input key lies.

After that it can use the vector defining the hyperplane H_i to test whether the input key could be an element of X or is surely not an element of X.

This construction of a key detection algorithm would work well if we could find the hyperplane of V partitioning X in approximately equal sized halves. The problem consists of the two following questions:

- How well can we "halve" the set X by a suitable hyperplane, or, equivalently, how well does the *optimal* hyperplane perform the partitioning of X?
- How to find this optimal hyperplane?

Chapter 2

The vector partition problem over $\mathbb{F}_2$

2.1 Introduction

In this chapter we present the basic problem, which can be stated as follows. Given a set of binary vectors, how well can we "halve" this set by intersecting it with a suitable hyperplane? Or, stated more precisely: given a subset X of a vector space V over $\mathbb{F}_2$, find a vector $a \in V \setminus \{0\}$ such that the hyperplane $H_a := \{v \in V | (v, a) = 0\}$ and its complement partition X in parts of about the same size. We will refer to this problem as the *vector partition problem over* $\mathbb{F}_2$.

We will first be interested in how well we can partition the set X in this way. The question of finding a vector a that realizes the optimal partition of X will be studied in Section 4.4.

In the next chapter we generalize this problem to its weighted version over a general field $\mathbb{F}_q$.

In Section 2.2 we present the precise problem statement. In Section 2.3 we shall convert the problem to a relating coding problem concerning the occurrence of weights close to $n/2$ in a binary linear code of length n. A solution of this coding theory problem is presented in Section 2.4 and will be discussed in Section 2.5.

2.2 Precise problem statement

Throughout this chapter, V will denote a fixed m-dimensional vector space over $\mathbb{F}_2$, that is, $V = \mathbb{F}_2^m$. We define $V^* = V \setminus \{0\}$. For $a \in V^*$, we will denote the hyperplane $\{v \in V | (v, a) = 0\}$ orthogonal to a in V by H_a. When we speak about the dimension of X, we will mean the dimension of the subspace spanned by X.

We desire to "halve" a given set $X \subseteq V$ as well as possible by intersecting X with some hyperplane H_a, that is, we are interested in the minimum

$$\delta(X) := \min_{a \in V} \Big| |H_a \cap X| - |X|/2 \Big|, \tag{2.1}$$

which measures how well we can do for this set X. We should not include the case $a = 0$, but for reasons of simplicity we still write V instead of V^*, which could by justified by defining $H_0 = V$. Note that the minimum will never be attained by $a = 0$.

We would like to obtain upperbounds on $\delta(X)$ given only the dimension m of V and the cardinality $|X| = n$ of the set $X \subseteq V$, i.e., we would like to obtain information on

$$f(m, n) := \max_{\substack{X \subseteq V \\ |X| = n}} \delta(X). \tag{2.2}$$

We will see below that the quantity $\delta(X)$ does not depend on dimension of the precise vector space V in which we embed X, but only on the dimension of X itself. Therefore, instead of carrying on with the function $f(m, n)$ we define

$$g(k, n) := \max_{\substack{\dim(X) = k \\ |X| = n}} \delta(X). \tag{2.3}$$

We now have the following.

Lemma 1 *With $f(m, n)$ and $g(k, n)$ defined as above, we have*

$$f(m, n) = \max_{0 \le k \le m} g(k, n). \tag{2.4}$$

Proof: Obviously, for any $X \subseteq V$ for which $f(m, n) = \delta(X)$, we have that $k = \dim(X) \le m$. Hence $f(m, n) \le g(k, n)$ for some k with $0 \le k \le m$.

Conversely, let X be such that $\dim(X) = k \leq m$ and $g(k,n) = \delta(X)$. Embed X in V by adding $m-k$ coordinates to the vectors from X with value zero. It's clear that this doesn't affect the quantity $\delta(X)$ nor the dimension of X. Hence $f(m,n) \geq \delta(X) = g(k,n)$ for all k with $0 \leq k \leq m$. □

Note that this lemma proves that $f(m,n)$ is non-decreasing in m. Furthermore, we may suppose without loss of generality that $0 \notin X$. Indeed, if we define $X+d$ for $d \in V$ as $\{x+d | x \in X\}$, then, except for the trivial case $X = V$, we can always find a $d \in V$ such that $0 \notin X+d$; the fact that this translation doesn't affect how well we can halve the set X by a hyperplane H_a, is stated in the following lemma.

Lemma 2 *We have*

$$\delta(X+d) = \delta(X) \text{ for all } d \in V. \tag{2.5}$$

Proof: For any hyperplane H_a in V and for all $d \in V$, we have

$$\begin{aligned} H_a \cap (X+d) &= \{x \in X+d | (a,x) = 0\} = \{x \in X | (a, x+d) = 0\} \\ &= \{x \in X | (a,x) + (a,d) = 0\} \\ &= \begin{cases} \{x \in X | (a,x) = 0\} = H_a \cap X & \text{if } (a,d) = 0, \\ \{x \in X | (a,x) = 1\} = X \backslash (H_a \cap X) & \text{if } (a,d) = 1. \end{cases} \end{aligned} \tag{2.6}$$

Now, by using the definition of $\delta(X)$ in (2.1), we can easily see that in both cases $\delta(X+d) = \delta(X)$ holds. □

2.3 A related coding problem

In this section, we will reformulate the vector partition problem in terms of coding theory. This will result in an alternative formulation of $\delta(X)$, in which the weights of codewords in a code corresponding to X play a prominent role.

Let $\mathrm{supp}(c)$ and $w(c)$ be the *support* and *Hamming-weight* of c, respectively, that is,

$$\mathrm{supp}(c) := \{i | c_i \neq 0\} \tag{2.7}$$

and

$$w(c) := |\mathrm{supp}(c)|. \tag{2.8}$$

Now let $X \subseteq \mathbb{F}_2^k$ with $n = |X|$ and $\dim(X) = k$ be the set that we want to partition. Let $G(X)$ be the $k \times n$-matrix which has as its columns the vectors from X. Note that $G(X)$ has full row rank. Let $\mathcal{C}(X)$ be defined as the binary $[n, k]$-code generated by $G(X)$, i.e., $\mathcal{C}(X)$ is the row space of $G(X)$. When no confusion can arise, we simply write $\mathcal{C}$ instead of $\mathcal{C}(X)$.

Recall that we assumed $0 \notin X$, so we have that the code $\mathcal{C}$ is *projective*: the columns of its generator matrix are pairwise linearly independent.

Now we are ready to write $\delta(X)$ in terms of the weights of the codewords from $\mathcal{C}(X)$ as stated in the following lemma.

Lemma 3 *We have*

$$\delta(X) = \min_{c \in \mathcal{C}(X)} |n/2 - w(c)|. \tag{2.9}$$

Proof: Let $c(a) = a^T G(X) \in \mathcal{C}$. Note that we may think of the coordinates of $c(a)$ as being indexed by the set $X = \{x_1, \ldots, x_n\}$, where the i-th coordinate $c_i(a)$ of $c(a)$ equals (a, x_i). As a consequence, we have

$$\begin{aligned} |H_a \cap X| &= |\{x \in X | (a, x) = 0\}| \\ &= |\{1 \leq i \leq n | c_i(a) = 0\}| \\ &= n - |\mathrm{supp}(c(a))| \\ &= n - w(c(a)). \end{aligned} \tag{2.10}$$

Hence

$$\begin{aligned} \delta(X) &= \min_{a \in V} ||H_a \cap X| - |X|/2| = \min_{a \in V} |n - w(c(a)) - n/2| \\ &= \min_{c \in \mathcal{C}(X)} |n/2 - w(c)|, \end{aligned} \tag{2.11}$$

where the last step is justified by the existence of a one-to-one correspondence between vectors $a \in \mathbb{F}_2^k$ and codewords $c \in \mathcal{C}$. □

Because of the correspondence between sets X with $\dim(X) = k$, $|X| = n$ and binary projective $[n, k]$-codes $\mathcal{C}$, we can write $g(k, n)$ as

$$g(k, n) = \max_{\mathcal{C}} \min_{c \in \mathcal{C}} |n/2 - w(c)|, \tag{2.12}$$

where the maximum is over all binary projective $[n, k]$-codes $\mathcal{C}$.

As a result we have reformulated the vector partition problem over $\mathbb{F}_2$ as a problem concerning the existence of codewords with weights close to $n/2$ in binary projective $[n,k]$-codes.

Before continuing we introduce some notation from coding theory. The *dual code* $C^\top$ of the binary $[n,k]$-code C is defined by

$$C^\top := \{v \in F_2^n | (c,v) = 0 \text{ for all } c \in C\}. \tag{2.13}$$

We use the symbols A_w and B_w to denote the number of codewords of weight w in C and $C^\top$, respectively. A and B are called the *weight distribution* of C and $C^\perp$.

2.4 A bound on $g(k,n)$

The next theorem concerns the weight distribution of binary projective $[n,k]$-codes and will directly lead to a bound on $g(k,n)$. This bound will be further discussed in the next section.

Theorem 1 *Let C be a binary projective $[n,k]$-code with weight distribution A. If $A_w = 0$ for all w that satisfy $n/2 - R < w < n/2 + R$, then*

$$R \leq \frac{1}{2}\sqrt{n - \frac{n^2-n}{2^k-1}}. \tag{2.14}$$

Equality holds if and only if C is a two-weight code with non-zero weights $n/2 \pm R$, that is, if and only if $A_w \neq 0$ implies that $w \in \{0, \frac{n-R}{2}, \frac{n+R}{2}\}$.

Proof: Let us define

$$N_j := \sum_{w=0}^{n} \binom{w}{j} A_w, \tag{2.15}$$

and write

$$w_1 = \frac{n}{2} - R, \qquad w_2 = \frac{n}{2} + R. \tag{2.16}$$

The proof of the theorem will consist of the following three steps.

1. First we will show that N_j, $0 \leq j \leq 2$, can be expressed in terms of k and n only. Since the polynomials $\binom{w}{j}$, $0 \leq j \leq 2$, form a basis for the space of polynomials of degree at most 2, it follows that all expressions $\sum_{w=0}^{n} p(w)A_w$ with a known polynomial p of degree at most 2 can be calculated explicitly.

2. Next, we consider the expression

$$E = \sum_{w=0}^{n} p(w)A_w, \tag{2.17}$$

where $p(w) = (w - w_1)(w - w_2)$.

Since p is of degree 2, we can use the results in step 1 to express E in terms of n and k only.

3. Finally, we will use our assumptions on the A_w to show that $E \geq w_1 w_2 = n^2/4 - R^2$. As a consequence, we obtain our desired bound on R.

The readers familiar with the MacWilliams Equations (see [9]) and the related Pless Power Moments (see [10]), will recognize step 1 as the computation of the first three "binomial moments" in this special case. The special property of the codes treated here is that they are projective, which implies that the minimal distance of their dual code is as least three, that is, the first three values of their dual weight distribution B are given by $B_0 = 1$, $B_1 = 0$, and $B_2 = 0$. The method of using the Pless Power Moments for such purposes originated from Kasami (see [6]). For a recent use of such a method, see [11].

Let us now turn to the details of the proof.

1. Note that since A_w counts the number of codewords of weight w, we may interpret the expression $\binom{w}{j}A_w$ as counting the number of pairs (S, c) with $c \in \mathcal{C}$, $w(c) = w$, $S \subseteq \mathrm{supp}(c)$ and $|S| = j$. As a consequence, with N_j as defined in (2.15), we have that

$$N_j = \sum_{w=0}^{n} \binom{w}{j} A_w = \sum_{c \in C} \sum_{\substack{S \subseteq \mathrm{supp}(c) \\ |S|=j}} 1 = \sum_{\substack{S \subseteq [1,n] \\ |S|=j}} |\mathcal{C}[S]|, \tag{2.18}$$

where we define

$$C[S] = \{c \in C | \mathrm{supp}(c) \supseteq S\}. \tag{2.19}$$

Since we assumed that C is projective, we can express $|C[S]|$ for $|S| \leq 2$ in terms of n and k only. Indeed, since any two columns in its generator matrix are independent, we immediately have that $|C[S]| = 2^{-|S|}|C| = 2^{k-j}$ when $|S| = j \leq 2$. So from (2.18), we now immediately have that

$$N_0 = 2^k, \qquad N_1 = n2^{k-1}, \qquad N_2 = \binom{n}{2}2^{k-2}. \tag{2.20}$$

2. Now consider the expression E in (2.17). We will use the expressions for N_j, $0 \leq j \leq 2$, in (2.20) to compute E. Since

$$\begin{aligned} p(w) &= (w - w_1)(w - w_2) \\ &= 2\binom{w}{2} - (w_1 + w_2 - 1)\binom{w}{1} + w_1 w_2 \binom{w}{0}, \end{aligned} \tag{2.21}$$

we have that

$$\begin{aligned} E &= 2N_2 - (w_1 + w_2 - 1)N_1 + w_1 w_2 N_0 \\ &= n(n-1)2^{k-2} - (n-1)n2^{k-1} + (\frac{n^2}{4} - R^2)2^k \\ &= (\frac{n^2}{4} - R^2)2^k - n(n-1)2^{k-2}. \end{aligned} \tag{2.22}$$

3. By assumption, $p(w)A_w = 0$ for $w_1 < w < w_2$, and obviously $p(w)A_w \geq 0$ for $0 \leq w \leq w_1$ or $w_2 \leq w \leq n$. Since $p(0)A_0 = w_1 w_2$, we immediately have that

$$E \geq w_1 w_2 = \frac{n^2}{4} - R^2. \tag{2.23}$$

Note that we have equality in (2.23) precisely when C is a two-weight code with non-zero weights w_1 and w_2.

Hence using (2.22), we find that

$$\frac{n^2}{4} - R^2 \leq (\frac{n^2}{4} - R^2)2^k - n(n-1)2^{k-2}, \tag{2.24}$$

and therefore

$$(2^k-1)R^2 \leq (2^k-1)\frac{n^2}{4} - n(n-1)2^{k-2}$$
$$= \frac{1}{4}(n2^k - n^2), \tag{2.25}$$

that is

$$R^2 \leq \frac{1}{4}(n - \frac{n^2-n}{2^k-1}). \tag{2.26}$$

□

From the above theorem, we immediately have the following.

Corollary 1 *We have that*

$$g(k,n) \leq \frac{1}{2}\sqrt{n - \frac{n^2-n}{2^k-1}}. \tag{2.27}$$

Equality holds if and only if there exists a binary two-weight $[n,k]$-code with weights lying symmetrically around $n/2$.

Proof: Suppose C is the binary $[n,k]$-code with the distinct column property for which the maximum in equation (2.12) is attained. Then

$$g(k,n) = \min_{c \in C} |n/2 - w(c)|. \tag{2.28}$$

Note that the restriction on A_w in Theorem 1 is equivalent to $\min_{c\in C}|n/2 - w(c)| > R$. So Theorem 1 now says the following: if $g(k,n) \geq R$ then

$$R \leq \frac{1}{2}\sqrt{n - (n^2-n)/(2^k-1)} \tag{2.29}$$

which is equivalent to the first statement of the corollary.

Theorem 1 further states that equality holds if and only if C is a two-weight code with nonzero weights $n/2 \pm R$, from which the second statement of the corollary follows immediately. □

In the sequel, we will refer to the bound in Corollary 1 as the *square root bound for $\mathbb{F}_2$*.

2.5 Discussion

In the introductory example (see Section 1.4) we already found a (necessarily two-weight) code C that attains the bound in Theorem 1. So this bound is sharp for at least some values of k and n. Later on, in section 4.2, we will present a family of two-weight codes with weights lying symmetrically around $n/2$, of which the mentioned example is the first one.

The bound presented here is non-decreasing in k. However, we can easily see that $g(n,n)$ must be equal to zero for even n and equal to $1/2$ for odd n. Indeed, a $[n,n]$-code is equal to the whole space $\mathbb{F}_2^n$ in which many words of weight $\lfloor n/2 \rfloor$ exist. So our bound is bad for values k close to n. However, in practice we usually have $k \ll n$.

Furthermore, in practical applications we often have $2^k \gg n$; in that case the bound is nearly equal to $1/2\sqrt{n}$, which shows that in such cases we have a pretty good bound. For example, if we take $k = 40$ and $n = 2000$, then $g(k,n) \leq 22$, hence, if we have a collection of 2000 40-bit vectors, we can "halve" the set taking the inner product of its elements with a single suitable 40-bit vector such that each half contains between 978 and 1022 elements.

Chapter 3

The weighted vector partition problem over $\mathbb{F}_q$

3.1 Introduction

In the previous chapter we showed that, given a set $X \subseteq \mathbb{F}_2^k$ of size $|X| = n$, there exists a vector $a \in \mathbb{F}_2^k$ such that the hyperplane H_a and its complement partition X in about two equal halves, and we obtained a precise bound of order $\sqrt{n}$ on how well this ideal can be approached.

Here we shall generalize that result to general fields $\mathbb{F}_q$, q prime power; moreover, instead of a given subset X of a k-dimensional vector space V over $\mathbb{F}_q$, we consider a "weight function" $\mu : V \to \mathbb{R}$ and we try to find a vector $a \in V$ such that the hyperplane H_a and its cosets each have approximately the same total weight. We will refer to this problem as the *weighted vector partition problem over* $\mathbb{F}_q$.

As it turns out, the techniques developed in the previous chapter can be generalized to deal with this situation in a similar way.

Our approach is the following. First we will show that we may assume that μ assigns non-negative integer weights to the vectors in V, that is, we may assume $\mu : V \to \mathbb{Z}_+$. Then, as in the previous chapter, we define a code $C(\mu)$, now of length $n = \sum_{v \in V} \mu(v)$. Finally, using similar ideas from coding theory, we can find an upperbound on the function that we are interested in.

The organization of this chapter will in general be the same as in the previous chapter. Section 3.2 contains the precise problem statement; in Section 3.3 we will show that we can assume that μ is a non-negative integer

weight function; this enables us to convert the problem to a related coding problem in Section 3.4. A solution of this problem will be given in Section 3.5 and will be discussed in Section 3.6.

3.2 Precise problem statement

Let the weight function $\mu : V \to \mathbb{R}$ be given, where V is a k-dimensional vector space over F_q. For $\lambda \in \mathbb{F}_q$, $a \in V \setminus \{0\}$, we define the coset $H_{a,\lambda}$ of the hyperplane H_a as

$$H_{a,\lambda} := \{v \in V | (v, a) = \lambda\}. \tag{3.1}$$

For all $X \subseteq V$, we also define

$$\mu(X) := \sum_{x \in X} \mu(x). \tag{3.2}$$

Given the weight function μ, we want to find the vector $a \in V$ that minimizes the sum of the quadratic differences between the total weight of the cosets of H_a and their desired values. We denote this minimum by $\delta(\mu)$; so we have

$$\delta(\mu) := \min_{a \in V} \sum_{\lambda \in \mathbb{F}_q} \left(\mu(H_{a,\lambda}) - \frac{1}{q}\mu(V)\right)^2. \tag{3.3}$$

Because $H_{0,\lambda}$ is not defined, we should not have $a = 0$ as a candidate for the partitioning of V, but for reasons of simplicity we still minimize also over $a = 0$. Note that, even if we define H_0 in a suitable way, the minimum will never be attained when $a = 0$.

Given an arbitrary weight function $\mu : V \to \mathbb{R}$, we are interested in finding a good upperbound on $\delta(\mu)$.

For exactly the same reason as in the previous chapter, we will only be interested in functions μ for which the dimension of its *support* $\{v \in V | \mu(v) \neq 0\}$, equals the dimension of V.

To illustrate this, suppose the dimension of the support of μ equals m with $m < k$. So there exists a linear subspace M, of dimension m, such that $m(v) \neq 0$ implies that $v \in M$. Then for any $W \subseteq V$, we have that $\mu(W) = \mu(W \cap M)$, from which follows that we could restrict the domain of μ from V to its linear subspace M with dimension m, without changing

$\delta(\mu)$. For a similar reason we worked in the previous chapterwith the function $g(k,n)$ instead of $f(m,n)$. Here we will simply assume from now on that the support of μ has a dimension equal to that of V.

3.3 Conversion to non-negative integer weights

In this section, we will show that we can assume that μ is a non-negative integer weight function. This will be achieved in three steps.

- It is clear that $\delta(\mu)$ is continuous in $\mu(v)$ for each $v \in V$. Suppose $\mu(x) = \alpha \in \mathbb{R}\backslash\mathbb{Q}$. Because of the continuity of $\delta(\mu)$, for each $\epsilon > 0$ we can find a $\delta > 0$ such that $\alpha + \delta \in \mathbb{Q}$ and $\delta(\mu) - \delta(\hat{\mu}) < \epsilon$, where $\hat{\mu}(v) = \mu(v)$ for $v \in V\backslash\{x\}$ and $\hat{\mu}(x) = \alpha + \delta$. Hereby we have shown that we can "replace" the function values where $\mu(x)$ is irrational by rational values, thus constructing a rational weight function $\hat{\mu}$ such that $\delta(\hat{\mu})$ is arbitrarily close to $\delta(\mu)$.

- If we define $\hat{\mu} := c\mu$, that is, $\hat{\mu}(v) = c\mu(v)$ for all $v \in V$, we have

$$\begin{aligned}\delta(\hat{\mu}) &= \min_{a\in V} \sum_{\lambda\in\mathbb{F}_q} \left(\hat{\mu}(H_{a,\lambda}) - \frac{1}{q}\hat{\mu}(V)\right)^2 \\ &= \min_{a\in V} \sum_{\lambda\in\mathbb{F}_q} \left(c\mu(H_{a,\lambda}) - \frac{1}{q}c\mu(V)\right)^2 \\ &= c\delta(\mu). \end{aligned} \tag{3.4}$$

- From the definition, it follows directly that $\delta(\mu)$ is invariant under translation of the weights. Indeed, if we take $\hat{\mu} = \mu + c$, that is, $\hat{\mu}(v) = \mu(v) + c$ for all $v \in V$, then

$$\begin{aligned}\delta(\hat{\mu}) &= \min_{a\in V} \sum_{\lambda\in\mathbb{F}_q} \left(\hat{\mu}(H_{a,\lambda}) - \frac{1}{q}\hat{\mu}(V)\right)^2 \\ &= \min_{a\in V} \sum_{\lambda\in\mathbb{F}_q} \left(\mu(H_{a,\lambda}) + |H_{a,\lambda}|c - \frac{1}{q}(\mu(V) + |V|c)\right)^2 \\ &= \min_{a\in V} \sum_{\lambda\in\mathbb{F}_q} \left(\mu(H_{a,\lambda}) + q^{k-1}c - \frac{1}{q}\mu(V) - q^{k-1}c\right)^2 \\ &= \delta(\mu). \end{aligned} \tag{3.5}$$

By applying these three steps we can "transform" the image of V under μ from $\mathbb{R}$ to $\mathbb{Q}$, then to $\mathbb{Z}$ and eventually to $\mathbb{Z}_+$. So from now on we will assume that μ is a non-negative integer weight function on V.

3.4 A related coding problem

In this section, we will convert the weighted vector partition problem over $\mathbb{F}_q$ to a problem formulated in terms of coding theory. We will construct a code $\mathcal{C}(\mu)$ from the given non-negative integer weight function μ, resulting in an alternative formulation of $\delta(\mu)$ in terms of generalized weights of the codewords in $\mathcal{C}(\mu)$.

For $\lambda \in F_q$, let $\mathrm{supp}_\lambda(c)$, $\mathbf{w}_\lambda(c)$, and $\mathbf{w}(c)$ be the *λ-support*, *λ-weight*, and *generalized weight* of c, respectively, defined as

$$\mathrm{supp}_\lambda(c) := \{i | c_i = \lambda\}, \tag{3.6}$$

$$\mathbf{w}_\lambda(c) := |\mathrm{supp}_\lambda(c)|, \tag{3.7}$$

$$\mathbf{w}(c) := (\mathbf{w}_\lambda(c))_{\lambda \in F_q}. \tag{3.8}$$

Furthermore, we write $[0, n] = \{0, \ldots, n\}$, and we use $\mathbf{1}$ and $\mathbf{e}_\lambda$, $\lambda \in \mathbb{F}_q$, to denote the all-one vector and the λ-th unit vector in $[0, n]^q$, respectively. Also, for a generalized weight $\mathbf{w} = \mathbf{w}(c)$, $c \in \mathcal{C}$, we let

$$\|\mathbf{w}\| := \sqrt{\sum_{\lambda \in \mathbb{F}_q} \mathbf{w}_\lambda^2}, \tag{3.9}$$

that is, $\|\mathbf{w}\|$ is the L_2-norm of the vector $\mathbf{w}$.

Finally, we let A denote the *generalized weight distribution* of a code $\mathcal{C}$ by defining

$$A_{\mathbf{w}} := \left|\{c \in \mathcal{C} | \mathbf{w}(c) = \mathbf{w}\}\right|. \tag{3.10}$$

Let $\mu : V = \mathbb{F}_q^k \to \mathbb{Z}_+$ be the weight function for which we want to find a hyperplane H_a such that the cosets of this hyperplane have approximately the same total weight. Let $n = \mu(V)$.

Now let $G(\mu)$ be the $k \times n$-matrix, which has the vectors v of V as its columns, each repeated $\mu(v)$ times. We define $\mathcal{C}(\mu)$ as the code generated by $G(\mu)$. When no confusing can arise, we will just write $\mathcal{C}$ instead of $\mathcal{C}(\mu)$.

Then $\mathcal{C}$ is a $[n,k]$-code over $\mathbb{F}_q$. Note that we still assume that the dimension of the support of μ equals k, so that $G(\mu)$ has full rank.

Now we are ready to write $\delta(\mu)$ in terms of the generalized weights of the codewords from the code $\mathcal{C}(\mu)$ as stated in the following lemma.

Lemma 4 *We have*

$$\delta(\mu) = \min_{c \in \mathcal{C}(\mu)} \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2. \tag{3.11}$$

Proof: Let $c(a) = a^\top G(\mu) \in \mathcal{C}(\mu)$. Then we have

$$\begin{aligned}\mu(H_{a,\lambda}) &= \sum_{v \in H_{a,\lambda}} \mu(v) = \sum_{\substack{v \in V \\ (a,v)=\lambda}} \mu(v) \\ &= \sum_{\substack{1 \le i \le n \\ (a,G(\mu)_i)=\lambda}} 1 = \big|\{1 \le i \le n | (a, G(\mu)_i) = c(a)_i = \lambda\}\big| \\ &= \mathbf{w}_\lambda(c(a)).\end{aligned} \tag{3.12}$$

Hence

$$\begin{aligned}\delta(\mu) &= \min_{a \in V} \sum_{\lambda \in \mathbb{F}_q} \big(\mu(H_{a,\lambda}) - \frac{1}{q}\mu(V)\big)^2 = \min_{a \in V} \sum_{\lambda \in \mathbb{F}_q} \big(\mathbf{w}_\lambda(c(a)) - \frac{n}{q}\big)^2 \\ &= \min_{c \in \mathcal{C}(\mu)} \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2,\end{aligned} \tag{3.13}$$

where the last step is justified by the existence of a one-to-one correspondence between the vectors $a \in V$ and codewords $c \in \mathcal{C}$. □

We conclude that the hyperplane H_a that partitions V best corresponds to the codeword c whose generalized weight lies closest to $\frac{n}{q}\mathbf{1}$.

3.5 An upperbound on $\delta(\mu)$

The next theorem gives an upperbound on $\delta(\mu)$ for arbitrary non-negative integer weight functions $\mu : V = \mathbb{F}_q^k \to \mathbb{Z}_+$ with support of dimension k. This bound will be further discussed in the next section.

Theorem 2 *For any non-negative integer weight function $\mu : \mathbb{F}_q^k \to \mathbb{Z}_+$ with support of dimension k, we have*

$$\delta(\mu) \leq \frac{(q-1)}{q} \frac{q^k}{(q^k-1)} \left(\sum_{v \in V} \mu^2(v) - \frac{1}{q^k} \left(\sum_{v \in V} \mu(v) \right)^2 \right) \tag{3.14}$$

Proof: For all $\mathbf{j} \in [0,n]^q$, we define

$$N_{\mathbf{j}} := \sum_{\mathbf{w}} \binom{\mathbf{w}}{\mathbf{j}} A_{\mathbf{w}}, \tag{3.15}$$

where $\binom{\mathbf{w}}{\mathbf{j}}$ is the generalized binomial coefficient, defined as

$$\binom{\mathbf{w}}{\mathbf{j}} := \prod_{\lambda \in \mathbb{F}_q} \binom{w_\lambda}{j_\lambda}. \tag{3.16}$$

The proof of the theorem will consist of the following three steps, which are essentially the same steps as in the proof of Theorem 1. We assume that μ is an arbitrary weight function with the properties as stated in the theorem, and we let $C = C(\mu)$, with generalized weight distribution A.

1. First we will show that we can express N_0, $\sum_{\lambda \in \mathbb{F}_q} N_{\mathbf{e}_\lambda}$, and $\sum_{\lambda \in \mathbb{F}_q} N_{2\mathbf{e}_\lambda}$ in terms of q, k, n and $\sum_{v \in V} \mu^2(v)$ only. Since the multivariate polynomials $\binom{\mathbf{w}}{\mathbf{j}}$ form a basis for the space of multivariate polynomials in the variables $\mathbf{w}_g l$, $\lambda \in \mathbb{F}_q$, it follows that certain expressions $\sum_{\mathbf{w} \in W} p(\mathbf{w}) A_{\mathbf{w}}$ with known multivariate polynomial p can be explicitly calculated.

2. Next, we consider the expression

$$E := \sum_{\mathbf{w}} p(\mathbf{w}) A_{\mathbf{w}}, \tag{3.17}$$

where

$$p(\mathbf{w}) := \left\| \mathbf{w} - \frac{n}{q} \mathbf{1} \right\|^2.$$

Using the results in step 1, we can express E in terms of q, k, n and $\sum_{v \in V} \mu^2(v)$ only.

3. Finally, we use Lemma 4 to obtain restrictions on the generalized weights of the codewords from C. We use these restrictions to show that $E \geq \frac{q-1}{q}n^2 + (q^k - 1)\delta(\mu)$. As a consequence, we obtain the desired bound on $\delta(\mu)$.

Let us now proceed to the details of the proof.

1. Note that since A_w counts the number of codewords of generalized weight w, we may interpret the expression $\binom{w}{j}A_w$ as counting the number of pairs (S, c) with $c \in C$, $w(c) = w$ and $S = (S_\lambda)_{\lambda \in \mathbb{F}_q}$ with $S_\lambda \subseteq \mathrm{supp}_\lambda(c)$ and $|S_\lambda| = j_\lambda$. As a consequence, with N_j defined in (3.15), we have that

$$N_j = \sum |C[(S_\lambda)_{\lambda \in \mathbb{F}_q}]|, \tag{3.18}$$

where the sum is over all $(S_\lambda)_{\lambda \in \mathbb{F}_q}$ with $S_\lambda \subseteq [1, n]$ and $|S_\lambda| = j_\lambda$, and where $C[(S_\lambda)_{\lambda \in \mathbb{F}_q}]$ is defined by

$$C[(S_\lambda)_{\lambda \in \mathbb{F}_q}] := \{c \in C | \mathrm{supp}_\lambda(c) \supseteq S_\lambda \text{ for all } \lambda \in \mathbb{F}_q\}. \tag{3.19}$$

Taking $j = 0$, we have

$$N_0 = |C[(\emptyset)_{\lambda \in \mathbb{F}_q}]| = |C| = q^k. \tag{3.20}$$

If we let $j = ae_\lambda$ for some integer $a \geq 1$, we have that

$$N_{ae_\lambda} = \sum_{\substack{S \subseteq [1,n] \\ |S| = a}} |\{c \in C | \mathrm{supp}_\lambda(c) \supseteq S\}|. \tag{3.21}$$

So we have that

$$\begin{aligned} \sum_{\lambda \in \mathbb{F}_q} N_{e_\lambda} &= \sum_{\lambda \in \mathbb{F}_q} \sum_{i=1}^{n} |\{c \in C | c_i = \lambda\}| \\ &= \sum_{i=1}^{n} \sum_{\lambda \in \mathbb{F}_q} |\{c \in C | c_i = \lambda\}| \\ &= \sum_{i=1}^{n} |C| = nq^k. \end{aligned} \tag{3.22}$$

In order to compute $\sum_{\lambda \in \mathbb{F}_q} N_{2\mathbf{e}_\lambda}$, note firstly that, given some set $\{i,j\} \subseteq [1,n]$, the following holds.

$$\begin{aligned} \left|\{c \in C | c_i = c_j\}\right| &= \left|\{a \in V | (a, G_i) = (a, G_j)\}\right| \\ &= \left|\{a \in V | (a, G_i - G_j) = 0\}\right| \\ &= \begin{cases} |V| = q^k & \text{if } G_i = G_j; \\ |H_{G_i - G_j}| = q^{k-1} & \text{otherwise.} \end{cases} \end{aligned} \tag{3.23}$$

Secondly, note that

$$\begin{aligned} \left|\{\{i,j\} \subseteq [1,n] | G_i = G_j\}\right| &= \sum_{v \in V} \binom{\mu(v)}{2} \\ &= \frac{1}{2}\left(\sum_{v \in V} \mu^2(v) - \sum_{v \in V} \mu(v)\right) \\ &= \frac{1}{2}\left(\sum_{v \in V} \mu^2(v) - n\right), \text{ and} \\ \left|\{\{i,j\} \subseteq [1,n] | G_i \neq G_j\}\right| &= \binom{n}{2} - \left|\{\{i,j\} \subseteq [1,n] | G_i = G_j\}\right| \\ &= \frac{1}{2}\left(n^2 - \sum_{v \in V} \mu^2(v)\right). \end{aligned}$$

Combining the above, we find that

$$\begin{aligned} \sum_{\lambda \in \mathbb{F}_q} N_{2\mathbf{e}_\lambda} &= \sum_{\lambda \in \mathbb{F}_q} \sum_{\{i,j\} \subseteq [1,n]} \left|\{c \in C | c_i = c_j = \lambda\}\right| \\ &= \sum_{\{i,j\} \subseteq [1,n]} \left|\{c \in C | c_i = c_j\}\right| \\ &= \frac{1}{2}\left(\sum_{v \in V} \mu^2(v) - n\right) q^k + \frac{1}{2}\left(n^2 - \sum_{v \in V} \mu^2(v)\right) q^{k-1} \\ &= \frac{1}{2} q^{k-1}\left(n^2 - qn + (q-1) \sum_{v \in V} \mu^2(v)\right). \end{aligned} \tag{3.24}$$

2. Now consider the expression E in (3.17). By using some manipulations,

we can write E in terms of N_0, $\sum_{\lambda\in\mathbb{F}_q} N_{e_\lambda}$, and $\sum_{\lambda\in\mathbb{F}_q} N_{2e_\lambda}$ as follows.

$$\begin{aligned}
E &= \sum_{\mathbf{w}\in W} p(\mathbf{w})A_{\mathbf{w}} \\
&= \sum_{\mathbf{w}\in W}\left[\sum_{\lambda\in\mathbb{F}_q}(\mathbf{w}_\lambda - \frac{n}{q})^2\right]A_{\mathbf{w}} \\
&= \sum_{\mathbf{w}\in W}\sum_{\lambda\in\mathbb{F}_q}[2\binom{\mathbf{w}_\lambda}{2} + (1-2\frac{n}{q})\binom{\mathbf{w}_\lambda}{1} + (\frac{n}{q})^2\binom{\mathbf{w}_\lambda}{0}]A_{\mathbf{w}} \\
&= \sum_{\mathbf{w}\in W}\sum_{\lambda\in\mathbb{F}_q}[2\binom{\mathbf{w}}{2\mathbf{e}_\lambda} + (1-2\frac{n}{q})\binom{\mathbf{w}}{\mathbf{e}_\lambda} + (\frac{n}{q})^2\binom{\mathbf{w}}{\mathbf{0}}]A_{\mathbf{w}} \\
&= 2\sum_{\lambda\in\mathbb{F}_q} N_{2e_\lambda} + (1-2\frac{n}{q})\sum_{\lambda\in\mathbb{F}_q} N_{e_\lambda} + q(\frac{n}{q})^2 N_0 \qquad (3.25)
\end{aligned}$$

Using (3.20), (3.22) and (3.24), we obtain

$$\begin{aligned}
E &= 2\frac{1}{2}q^{k-1}(n^2 - qn + (q-1)\sum_{v\in V}\mu^2(v)) + (1-2\frac{n}{q})nq^k + q(\frac{n}{q})^2 q^k \\
&= (q-1)q^{k-1}\sum_{v\in V}\mu^2(v). \qquad (3.26)
\end{aligned}$$

3. By Lemma 4, we have that

$$\|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2 \geq \delta(\mu) \text{ for all } c \in C. \qquad (3.27)$$

We know that $0 \in C$ and we have $p(\mathbf{w}(0)) = p(n\mathbf{e}_0) = n^2(q-1)/q$. As a consequence,

$$\begin{aligned}
E &= \sum_{\mathbf{w}\in W} p(\mathbf{w})A_{\mathbf{w}} = \sum_{c\in C} p(\mathbf{w}(c)) \\
&= p(\mathbf{w}(0)) + \sum_{c\in C\setminus\{0\}} \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2 \\
&\geq \frac{q-1}{q}n^2 + (q^k - 1)\delta(\mu). \qquad (3.28)
\end{aligned}$$

Hence by (3.26), we find that

$$\frac{q-1}{q}n^2 + (q^k-1)\delta(\mu) \leq (q-1)q^{k-1}\sum_{v\in V}\mu^2(v) \qquad \Leftrightarrow$$

$$(q^k-1)\delta(\mu) \leq \frac{q-1}{q}(q^k\sum_{v\in V}\mu^2(v) - n^2). \qquad (3.29)$$

Recall that we defined $n = \sum_{v\in V}\mu(v)$. After this substitution the statement of the theorem immediately follows.

□

We will refer to the bound in Theorem 2 as the *square root bound for* $\mathbb{F}_q$.

3.6 Discussion

In this section we first will show that the "new bound", that is, the upper-bound on $\delta(\mu)$ found here, is in fact the same as the "old bound", that is, the bound found on $\delta(X)$ in the previous chapter. After that, we will show that the bound on $\delta(\mu)$ presented here can be written as a constant times the *variance* of μ. We will also look at a small example.

First, we will show that the new bound of this chapter indeed generalizes the old bound in the previous chapter, that is, the new bound reduces to the old bound in the case where $q = 2$ and $\mu : V = \mathbb{F}_2^k \to \{0,1\}$ is the indicator function μ_X of the set $X \subseteq V$ with $|X| = n$. The indicator function μ_X is defined by $\mu_X(x) = 1$ if $x \in X$, $\mu_X(x) = 0$ otherwise.

Now we have $n = |X| = \mu(V) = \sum_{v\in V}\mu(v) = \sum_{v\in V}\mu^2(v)$, so the new bound is in this case

$$\delta(\mu_X) \leq \frac{1}{2}\frac{2^k}{(2^k-1)}\left(n - \frac{1}{2^k}n^2\right)$$

$$= 2\left(\frac{1}{2}\sqrt{n - \frac{n^2-n}{2^k-1}}\right)^2. \qquad (3.30)$$

Note that we have that $\mu_X(H_{a,0}) = |H_a \cap X|$ and $\mu_X(H_{a,1}) = |X| - |H_a \cap X|$. With use of the definitions of $\delta(\mu_X)$ and $\delta(X)$, we now have that

$$\delta(\mu_X) = \min_{a\in V} 2\big(|H_a \cap X| - |X|/2\big)^2 = 2\delta^2(X). \qquad (3.31)$$

Substituting (3.31) in (3.30) gives us the old bound as follows.

$$2\delta^2(X) \leq 2\left(\frac{1}{2}\sqrt{n - \frac{n^2-n}{2^k-1}}\right)^2,$$

so that

$$\delta(X) \leq \left(\frac{1}{2}\sqrt{n - \frac{n^2-n}{2^k-1}}\right). \tag{3.32}$$

Next, we show that the bound on $\delta(\mu)$ that we have found in this chapter is a linear function of the variance $\sigma^2(\mu)$ of μ. As usual, we define $\sigma^2(\mu)$ as

$$\sigma^2(\mu) := \frac{1}{|V|}\sum_{v\in V}(\mu(v) - \bar{\mu})^2 \text{ where} \tag{3.33}$$

$$\bar{\mu} := \frac{1}{|V|}\sum_{w\in V}\mu(w). \tag{3.34}$$

It is easily seen that

$$\sigma^2(\mu) = \frac{1}{q^k}\left(\sum_{v\in V}\mu^2(v) - \frac{1}{q^k}\left(\sum_{v\in V}\mu(v)\right)^2\right). \tag{3.35}$$

Substituting the above expression in the bound of Theorem 2, we obtain

$$\delta(\mu) \leq \frac{(q-1)}{q}\frac{q^{2k}}{(q^k-1)}\sigma^2(\mu). \tag{3.36}$$

This shows that the upperbound on how well we can partition the weighted vector space V with a hyperplane H_a under the weight function μ depends in a linear way on how the weights are distributed. If all weights are equal we obtain an upperbound of zero, while weights that are arbitrarily spread out lead to an arbitrarily high upperbound.

Let's have a look at an fairly randomly chosen example for which $q = 8$, $k = 40$ and a weight function that takes 1000 times the value 1 and 1000 times the value 2 and equals zero for the rest of the space $V = \mathbb{F}_8^{40}$. Ideally we partition V in 8 subsets with each total weight $(1000 + 2 \cdot 1000)/8 = 375$. Theorem 2 then gives us the bound $\delta(\mu) \leq 4375$. If this bound would be

attained, then this means that the total weight of each subset would each have an *average* distance of about $\sqrt{4375/8} \approx 23$ to its desired value of 375. The maximum distance of the total weight of one subset to 375 would be 46 (which can only happen if six of the other seven subsets would have the ideal total weight).

Chapter 4

Other results

4.1 Introduction

...

4.2 A construction of sets attaining the square-root bound for $\mathbb{F}_2$

In this section we will show a construction of sets X for which the square-root bound for $\mathbb{F}_2$, stated in Corollary 1, is attained. This proves that this bound is sharp for certain values for k and n. We also will consider strongly regular graphs corresponding to two-weight codes.

First we will present a lemma which enables us to give a construction of sets X attaining the square-root bound for $\mathbb{F}_2$. This lemma gives the maximum number of m-dimensional linear subspaces of an mk-dimensional vector space that intersect trivially, that is, intersect in $\{0\}$ only).

Lemma 5 *The maximum number of k-dimensional linear subspaces V_i of $\mathbb{F}_q^{mk}$ for which $V_i \cap V_j = \{0\}$ if $i \neq j$ equals $(q^{mk}-1)/(q^k-1)$.*

Proof: The field $\mathbb{F}_q^{mk}$ contains $\mathbb{F}_q^k$ as a subfield, and $\mathbb{F}_q^k$ contains $\mathbb{F}_q$ as a subfield. For $x, y \in \mathbb{F}_q^{mk}$, define

$$x \sim y \quad \text{if and only if} \quad x = \lambda y \text{ for some } \lambda \in \mathbb{F}_q^k \setminus \{0\}. \tag{4.1}$$

We claim that $\sim$ is an equivalence relation. Indeed, obviously, if $x, y, z \in \mathbb{F}_q^{km}$, then $x \sim y$ (reflexive), if $x \sim y$ then $y \sim x$ (symmetric), and if $x \sim y$, $y \sim z$ then $x \sim z$ (transitive).

Let $E_0 = \{0\}, E_1, \ldots, E_R$ denote the equivalence classes of n, and write $V_i = E_i \cup \{0\}$.

We claim that V_i is an $\mathbb{F}_q$-linear subspace of dimension k. To see this, let $a \in E_i$. Then

$$V_i = \{\lambda a | \lambda \in \mathbb{F}_q^k\} \tag{4.2}$$

is obviously $\mathbb{F}_q$-linear. Also, $|V_i| = |\mathbb{F}_q^k| = q^k$, so as an $\mathbb{F}_q$-linear space, we have $\dim_{\mathbb{F}_q}(V_i) = k$.

For $i \geq 1$, we have $|E_i| = q^k - 1$ and we already had $|\bigcup_{i=1}^R E_i| = |\mathbb{F}_q^{mk} \setminus \{0\}| = q^{mk} - 1$, so now follows $R = (q^{mk} - 1)/(q^k - 1)$. □

The following lemma shows us how to construct sets $X \subseteq \mathbb{F}_q^{2k}$ for which the intersection of X with a hyperplane has only two possible cardinalities.

Lemma 6 *Let $V = \mathbb{F}_q^{2k}$ and $(V_i)_{1 \leq i \leq R}$ be a collection of k-dimensional linear subspaces of V for which $V_i \cap V_j = \{0\}$ for all i, j with $i \neq j$. If*

$$X = \bigcup_{i=1}^{R} V_i \setminus \{0\}, \tag{4.3}$$

then, for any hyperplane H of V,

$$|X \cap H| \in \{(R-1)(q^{k-1} - 1) + q^k - 1, R(q^{k-1} - 1)\}. \tag{4.4}$$

Proof: Let V, $(V_i)_{1 \leq i \leq R}$ and X be as in the lemma. Let H be an arbitrary hyperplane of V. Then we have that

$$|X \cap H| = \sum_{i=1}^{R} (|V_i \cap H| - 1), \tag{4.5}$$

and

$$\dim(V_i \cap H) = \begin{cases} \dim(V_i) & \text{if } V_i \subseteq H; \\ \dim(V_i) - 1 & \text{otherwise,} \end{cases} \tag{4.6}$$

from which follows that

$$|V_i \cap H| = \begin{cases} q^k & \text{if } V_i \subseteq H; \\ q^{k-1} & \text{otherwise.} \end{cases} \tag{4.7}$$

Now we claim that H contains at most one of the V_i's. Indeed, if $i \neq j$, then $V_i \cap V_j = \{0\}$, hence $V_i \cup V_j$ spans V.

By substituting (4.7) in (4.5), we obtain that

$$|X \cap H| = \begin{cases} (R-1)(q^{k-1}-1) + q^k - 1 & \text{if } V_i \subseteq H \text{ for some i;} \\ R(q^{k-1}-1) & \text{otherwise.} \end{cases} \tag{4.8}$$

□

Now, by taking $q = 2$ and picking the right value for R in the previous lemma, we can construct a set $X \subseteq \mathbb{F}_2^{2k}$ such that $\delta(X) = g(2k, n)$. Indeed, we have the following.

Theorem 3 *Let $q = 2$, $k \geq 1$, and let $V_1, \ldots, V_R$ and X be as in Lemma 6. Take $R := 2^{k-1}$. Then, writing $n = |X|$, we have that*

$$\delta(X) = 2^{k-2} = \frac{1}{2}\sqrt{n - \frac{n^2 - n}{2^{2k} - 1}}. \tag{4.9}$$

Proof: Let V, $V_1, \ldots, V_R$, X and n be defined as above. Note that, since $|V_i \setminus \{0\}| = 2^k - 1$, we have that n equals $R(2^k - 1) = 2^{k-1}(2^k - 1)$. Note also, that as a consequence of Lemma 5 there exist $(2^{2k} - 1)/(2^k - 1) = 2^k + 1$ k-dimensional subspaces of V with the desired property, so we may take $R = 2^{k-1}$.

Application of the definition of $\delta(X)$ and Lemma 6 gives us that

$$\begin{aligned} \delta(X) &= \min_{a \in V} \Big| |H_a \cap X| - |X|/2 \Big| \\ &= \min\big(\big|(R-1)(2^{k-1}-1) + 2^k - 1 - n/2\big|, \big|R(2^{k-1}-1) - n/2\big|\big) \\ &= \min(2^{k-2}, 2^{k-2}) \\ &= 2^{k-2}. \end{aligned} \tag{4.10}$$

□

Remark that, in order to have the equality $\delta(X) = g(2k, n)$ we want $\dim(X) = 2k$, so we need $R \geq 2$, that is, $k \geq 2$.

As a consequence, the code $C(X)$ obtained from a set X constructed in this way is a binary projective two-weight $[n, 2k]$-code with weights lying symmetrically around $n/2$. The length n of the code constructed in this way necessarily equals $2^{k-1}(2^k - 1)$ and the non-zero weights are $n/2 \pm 2^{k-2}$.

Now the question arises whether there exist any other codes attaining the square-root bound. If not, then we can look for a sharper bound on $g(k, n)$ for the pairs (k, n) not covered by the construction above, if so, then the question is how to find those codes.

Also from a coding theoretic point of view the existence of two-weight projective codes with weights lying symmetrically around $n/2$, or even two-weight codes in general, is an interesting topic. Note that, by picking an alternative value for R in the above theorem, we can "shift" the center of the two non-zero weights of the code, thus creating two-weight codes with weights *not* lying symmetrically around $n/2$.

Maybe we can either exclude or prove the existence of two-weight projective codes with certain parameters by looking at their corresponding *strongly regular graphs*. Let us first give the definition of such a graph. A strongly regular graph $srg(v, k, \lambda, \mu)$ is a graph with v vertices that is regular of degree k and that has the following properties:

1. For any two adjacent vertices x, y, there are exactly λ vertices adjacent to x and to y.

2. For any two nonadjacent vertices x, y, there are exactly μ vertices adjacent to x and to y.

In [2], Delsarte gave a construction of strongly regular graphs from two-weight projective codes. He also proved the existence of two-weight codes corresponding to certain strongly regular graphs. From his work, we can derive the relations between the parameters of two-weight codes (length, dimension, weight distribution) and the parameters of the corresponding strongly regular graphs (v,k,λ and μ as in the above definition and the eigenvalues and their multiplicities of the incidence-matrix of the graph).

The literature (see, for example, [1] and [8]) provides us with several theorems on the properties of the parameters of strongly regular graphs. For example, we have the *integrality condition* and the *Krein conditions*. At first sight, some of these results seem useful for our purposes, but up to now, we did not succeed by these methods to prove the (non-)existence of (families of)

two non-zero weight codes for which their (non-)existence was not formerly known.

4.3 Two ways of partitioning $\mathbb{F}_q^{km}$

In this section we consider the following problem. Suppose we are given a weight function $\mu : V \to \mathbb{R}$ where $V = \mathbb{F}_q^{mk}$. We wish to partition the space V into q^m parts, where each part has total weight of approximately $\mu(V)/q^m$ and where we use only linear functions to achieve the partition. We can think of two ways to produce such a partition.

1. Consider V as a k-dimensional vector space over $\mathbb{F}_{q^m}$, and as partition, take the q^m cosets of a hyperplane in $\mathbb{F}_{q^m}$. Then we can use the square-root bound to obtain a guarantee on how well we can do.

2. Consider V as a km-dimensional vector space over $\mathbb{F}_q$. In a first stage, partition V into q parts $V_1, \ldots, V_q$, consisting of the q cosets of a hyperplane in $\mathbb{F}_q^{km}$. Then, we repeat this procedure in each V_i (note that we may consider V_i as a $(km-1)$-dimensional vector space over $\mathbb{F}_q$), etc.. Again we can use the square-root bound to give a guarantee about how well we can do in each partition step.

Now the question is for which partition method our bounds produce the best overall upperbound.

4.4 Algorithms performing the partitioning

In Chapter 3, we investigated how well the cosets of a suitable hyperplane could partition V such that all cosets have approximately the same total weight.

In this section we investigate how to find a vector $a \in V$ that produces a good partition of V. We will show that a so-called *randomized algorithm* performs well. A randomized algorithm is an algorithm that randomly chooses vectors from V and outputs the vector that produces the best result.

Let $\mu : V = \mathbb{F}_q^* \to \mathbb{R}$ be the weight function on V for which we want to find a vector a that minimizes

$$\delta_a(\mu) := \sum_{\lambda \in \mathbb{F}_q} \left(\mu(H_{a,\lambda}) - \frac{1}{q}\mu(V)\right)^2. \tag{4.11}$$

Let $V \setminus \{0\}$ be denoted by V^*. By Theorem 2, there exists a vector $a \in V^*$ for which $\delta_a(\mu)$ is at most $B(\mu)$, where

$$B(\mu) := \frac{(q-1)}{q} \frac{q^k}{(q^k-1)} \left(\sum_{v \in V} \mu^2(v) - \frac{1}{q^k} \left(\sum_{v \in V} \mu(v) \right)^2 \right). \tag{4.12}$$

We would be satisfied if, for any weight function μ, we find an algorithm that always produces a vector a such that $\delta_a(\mu) \leq \theta B(\mu)$, for some θ. Therefore, we qualify a to be *θ-nice* if $\delta_a(\mu) \leq \theta B(\mu)$ holds. Note that, the lower the value of θ, the better the performance of such an algorithm. Note also that if we could find an algorithm that guarantees this for $\theta < 1$, then our bound would not be the minimal upperbound on $\delta_a(\mu)$.

The next theorem essentially states that, for any $\theta > 1$, a randomly chosen vector $a \in V^*$ has a positive probability of being θ-nice. It states a lower bound for this probability in terms of θ. Here we assume that we always choose *uniformly at random*, that is, each vector $a \in V^*$ has equal probability to be chosen.

Theorem 4 *Let $\mu : V = \mathbb{F}_q^k \to \mathbb{R}$ be a function with support of dimension k, and let $\theta > 1$. Then we have*

$$\text{Prob}(a \textit{ is } \theta\textit{-nice}) \geq \frac{\theta - 1}{\theta}. \tag{4.13}$$

Proof: Let C be the $[n,k]$-code $C(\mu)$ over $\mathbb{F}_q$ with generator matrix G and generalized weight distribution A. Since we choose $a \in V^*$ uniformly at random, we have that

$$\text{Prob}(a \text{ is } \theta\text{-nice}) = \frac{|\{a \in V^* | \delta_a(\mu) \leq \theta B(\mu)\}|}{|V^*|}. \tag{4.14}$$

In the proof of Lemma 4, we saw that

$$\delta_a(\mu) = \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2, \tag{4.15}$$

where c is the codeword corresponding to a, that is, $c(a) = a^T G$. Hence

$$|\{a \in V^* | \delta_a(\mu) \leq \theta B(\mu)\}| = |C_1|, \tag{4.16}$$

where

$$C_1 := \left\{ c \in C \setminus \{0\} \,\middle|\, \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2 \leq \theta B(\mu) \right\}. \tag{4.17}$$

Define

$$E := \sum_{\mathbf{w}} p(\mathbf{w}) A_{\mathbf{w}}, \tag{4.18}$$

where $p(\mathbf{w}) = \|\mathbf{w} - \frac{n}{q}\mathbf{1}\|^2$. As in the proof of Theorem 2, we have that

$$E = (q-1)q^{k-1} \sum_{v \in V} \mu^2(v), \tag{4.19}$$

and secondly we have that

$$\begin{aligned} E &= \sum_{c \in C} \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2 \\ &= \|\mathbf{w}(0) - \frac{n}{q}\mathbf{1}\|^2 + \underbrace{\sum_{c \in C_1} \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2}_{\geq 0} + \underbrace{\sum_{c \in (C \setminus C_1) \setminus \{0\}} \|\mathbf{w}(c) - \frac{n}{q}\mathbf{1}\|^2}_{\geq \left|(C \setminus C_1) \setminus \{0\}\right| \theta B(\mu)} \\ &\geq \frac{q-1}{q} n^2 + (q^k - |C_1| - 1)\theta B(\mu). \end{aligned} \tag{4.20}$$

By substituting (4.12) and solving for $|C_1|$, we find $|C_1| \geq (q^k - 1)(\theta - 1)/\theta$.

Combining this result with (4.16), (4.14), and $|V^*| = q^k - 1$, we obtain the statement in the theorem. □

Now consider the following algorithm: pick N times a vector from V^*, uniformly at random, and choose the vector that produces the best partition of V.

From the above theorem it immediately follows that the probability that *none* of the N vectors is θ-nice is less than $(1 - (\theta - 1)/\theta)^N = 1/\theta^N$. So we conclude that for $\theta > 1$ the failure-probability, that is, the probability that output of the algorithm is not θ-nice, goes to zero exponentially fast with increasing N.

4.5 Hadamard-matrix problem description

We can write the weighted vector partition problem over $\mathbb{F}_q$ also in terms of *generalized Hadamard-matrices*. An $m \times m$ Hadamard-matrix $\mathcal{H}$ is a $(-1, 1)$-matrix for which $\mathcal{H}^{\perp}\mathcal{H} = mI_m$ holds, where I_m denotes the $m \times m$ identity matrix. For later use, for any proposition P we define the function δ_P as

$$\delta_P := \begin{cases} 1 & \text{if } P \text{ is true;} \\ 0 & \text{if } P \text{ is false.} \end{cases} \tag{4.21}$$

Let $V = \mathbb{F}_q^k$. For λ in $\mathbb{F}_q$, we define the generalized $|V| \times |V|$ Hadamard-matrix $\mathcal{H}_\lambda$, indexed by the elements of V, by

$$\mathcal{H}_\lambda(a, b) := \delta_{(a,b)=\lambda} - \frac{1}{q}. \tag{4.22}$$

Note that for $q = 2$, the matrix $2\mathcal{H}_0$ is an ordinary Hadamard-matrix.

Let $\mu : V \to \mathbb{R}$ be the weight function on V. Let $\mu_v := \mu(v)$ for all $v \in V$. By a slight abuse of notation, we will also use μ to denote the vector $(\mu_v)_{v \in V}$ in $\mathbb{R}^V$.

Now we have that

$$\begin{aligned} (\mathcal{H}_\lambda \mu)_a &= \sum_{v \in V} \mathcal{H}_\lambda(a, v)\mu_v = \sum_{v \in V} \mu_v(\delta_{(a,v)=\lambda} - \frac{1}{q}) \\ &= \mu(H_{a,\lambda}) - \frac{\mu(V)}{q}, \end{aligned} \tag{4.23}$$

from which, using the definition in (3.3), we immediately obtain that

$$\delta(\mu) = \min_{a \in V} \sum_{\lambda \in \mathbb{F}_q} (\mathcal{H}_\lambda \mu)_a^2. \tag{4.24}$$

We see that finding the vector a that optimal partitions V is equivalent to finding the smallest quadratic sum over λ of the a-th component of the vectors $\mathcal{H}_\lambda \mu$. Now that we have given an alternative formulation for $\delta(\mu)$, we can prove Theorem 2 as follows.

Define

$$E := \sum_{\lambda \in \mathbb{F}_q} \|\mathcal{H}_\lambda \mu\|^2. \tag{4.25}$$

In order to calculate E we first prove that $\sum_{\lambda\in\mathbb{F}_q}\mathcal{H}_\lambda^\perp\mathcal{H}_\lambda=(q^k-q^{k-1})I_{|V|}$ by showing that $\big(\sum_{\lambda\in\mathbb{F}_q}H_\lambda^\perp\mathcal{H}_\lambda\big)(a,b)=(q^k-q^{k-1})\delta_{a=b}$. We have that

$$\begin{aligned}
\Big(\sum_{\lambda\in\mathbb{F}_q}H_\lambda^\perp\mathcal{H}_\lambda\Big)(a,b)&=\sum_{\lambda\in\mathbb{F}_q}\sum_{v\in V}\mathcal{H}_\lambda(a,v)\mathcal{H}_\lambda(b,v)\\
&=\sum_{v\in V}\sum_{\lambda\in\mathbb{F}_q}\Big(\delta_{(a,v)=\lambda}-\frac{1}{q}\Big)\Big(\delta_{(b,v)=\lambda}-\frac{1}{q}\Big)\\
&=\sum_{v\in V}\sum_{\lambda\in\mathbb{F}_q}\delta_{(a,v)=\lambda}\delta_{(b,v)=\lambda}-\frac{1}{q}\sum_{v\in V}\sum_{\lambda\in\mathbb{F}_q}\delta_{(a,v)=\lambda}-\frac{1}{q}\sum_{v\in V}\sum_{\lambda\in\mathbb{F}_q}\delta_{(b,v)=\lambda}+\frac{1}{q^2}\sum_{v\in V}\sum_{\lambda\in\mathbb{F}_q}1\\
&=\sum_{v\in V}\sum_{\lambda\in\mathbb{F}_q}\delta_{(a,v)=\lambda}\delta_{(a-b,v)=0}-\frac{1}{q}q^k-\frac{1}{q}q^k+\frac{1}{q^2}q^kq\\
&=\sum_{v\in V}\delta_{(a-b,v)=0}-q^{k-1}-q^{k-1}+q^{k-1}\\
&=q^k\delta_{a=b}+q^{k-1}\delta_{a\neq b}-q^{k-1}\\
&=(q^k-q^{k-1})\delta_{a=b}.
\end{aligned}\tag{4.26}$$

Here we used that $\sum_{\lambda\in\mathbb{F}_q}\delta_{(u,v)=\lambda}=1$, for all $u\in V$, and that $\delta_{a\neq b}=1-\delta_{a=b}$.

Let us now return to the calculation of E. We now have that

$$\begin{aligned}
E&=\sum_{\lambda\in\mathbb{F}_q}(\mathcal{H}_\lambda\mu,\mathcal{H}_\lambda\mu)=\sum_{\lambda\in\mathbb{F}_q}\mu^\perp\mathcal{H}_\lambda^\perp\mathcal{H}_\lambda\mu\\
&=\mu^\perp\Big(\sum_{\lambda\in\mathbb{F}_q}\mathcal{H}_\lambda^\perp\mathcal{H}_\lambda\Big)\mu=(q^k-q^{k-1})\sum_{v\in V}\mu_v^2,
\end{aligned}\tag{4.27}$$

where we used that $\mu^\perp\mu=\sum_{v\in V}\mu_v^2$.

By using the definition of $\|\cdot\|$, we have that

$$\begin{aligned}
E&=\sum_{\lambda\in\mathbb{F}_q}\sum_{v\in V}(\mathcal{H}_\lambda\mu)_a^2=\sum_{v\in V}\sum_{\lambda\in\mathbb{F}_q}(\mathcal{H}_\lambda\mu)_a^2\\
&\geq\sum_{\lambda\in\mathbb{F}_q}(\mathcal{H}_\lambda\mu)_0^2+(q^k-1)\delta(\mu)\\
&=\frac{q-1}{q}\Big(\sum_{v\in V}\mu_v\Big)^2+(q^k-1)\delta(\mu).
\end{aligned}\tag{4.28}$$

Now we can combine (4.27) and (4.28); we thus obtain that

$$(q^k - 1)\delta(\mu) \leq \frac{q-1}{q}(q^k \sum_{v \in V} \mu_v^2 - (\sum_{v \in V} \mu_v)^2), \qquad (4.29)$$

which is equivalent to the square root bound for $\mathbb{F}_q$ as stated in Theorem 2.

Bibliography

[1] P.J. Cameron, J.H. van Lint, *Designs, Graphs, Codes and their Links*, Cambridge University Press, 1991.

[2] Ph. Delsarte, *Weights of linear codes and storngly regular normed spaces*, Discrete Mathematics 3 (1972), 47–64.

[3] Delsarte ...

[4] M. van Eupen, *Ternary Linear Codes*, Ph.D. thesis, Eindhoven University of Technology, 1996, Chapter 5.

[5] L. Gillman, *Writing Mathematics Well*, The Mathematical Association of America, 1987.

[6] Kasami ...

[7] J.H. van Lint, *Lecture Notes in Mathematics, Coding Theory*, Springer-Verlag Berlin - Heidelberg, 1973.

[8] J.H. van Lint, R.M. Wilson, *A Course in Combinatorics*, Cambridge University Press, 1992.

[9] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, North-Holland, 1977.

[10] V. Pless, *Power Moment Identites on Weight Distributions in Error Correcting Codes*, Information en Control 6 (1963), 147–152.

[11] Welch ...

CLAIMS:

1. An apparatus for reading out information on an information carrier which apparatus refuses play back of the information carrier if a predefined condition, substantially as described above, is not matched.

2. An apparatus according to claim 1, wherein the apparatus is a DVD-player.

3. An apparatus according to claim 1 or 2, wherein the predefined condition is the presence of a physical disc mark present on the information carrier upon detection of a trigger, substantially as described above.

4. An apparatus according to claim 3, wherein the physical disc mark is a wobble.

5. An apparatus according to claim 4, wherein the trigger is a wobble trigger, substantially as described above.

6. An information carrier comprising a physical disc mark and a trigger, substantially as described herein.

7. An information carrier according to claim 6, wherein the trigger is a wobble trigger, substantially as described above.

8. An information carrier according to claim 7, wherein the trigger is a single bit trigger.

9. An information carrier according to claim 7, wherein the wobble trigger is encoded in a pattern of encrypted and unencrypted packs.

10. An information carrier according to claim 6, wherein the physical disc mark is a wobble.

11. An information carrier according to claim 6, 7, 8, 9 or 10, wherein the information carrier is a DVD-disc.

12. An information carrier according to claim 7, wherein the wobble trigger is encoded in a key detection algorithm, which algorithm is used to detect whether the information carrier contains "old" or "new" content.

13. A method of copy protection of content present on an information carrier substantially as described herein.

14. A method of exchanging copy protection information regarding an information carrier substantially as described herein.

15. A copy protection system substantially as described herein.

ABSTRACT:

The invention tries to find a realisation of the CSS-rule: *CSS encrypted content on a recordable disc should be refused.* In order to be able to use a wobbled disc for distinguishing ROM-discs from recordable discs, it is required that *in the content* on "new" discs there will be a "wobble-trigger". This trigger has the following requirements: - it should be easily detectable from looking just at the content, -it should not be easily removable by a hacker, - it should not affect content preparation.

Two solutions for this wobble trigger are proposed. The *first proposed solution* is based on the recognition that a message for the purpose of copy protection may be transmitted by the deliberately encrypting packs following a certain pattern. Something like pseudo-random noise patterns of unencrypted packs and encrypted packs would be more suitable. The *second proposed solution* consists of designing a function operating on the key $K: \rightarrow f(K)$, where $f(K)$ can be 0 or 1. $f()$ has to be chosen in such a way that when operating on the keys used in the DVD-titles published so far (on the order of 4000 keys), it always yields 0.

(Fig. 2)

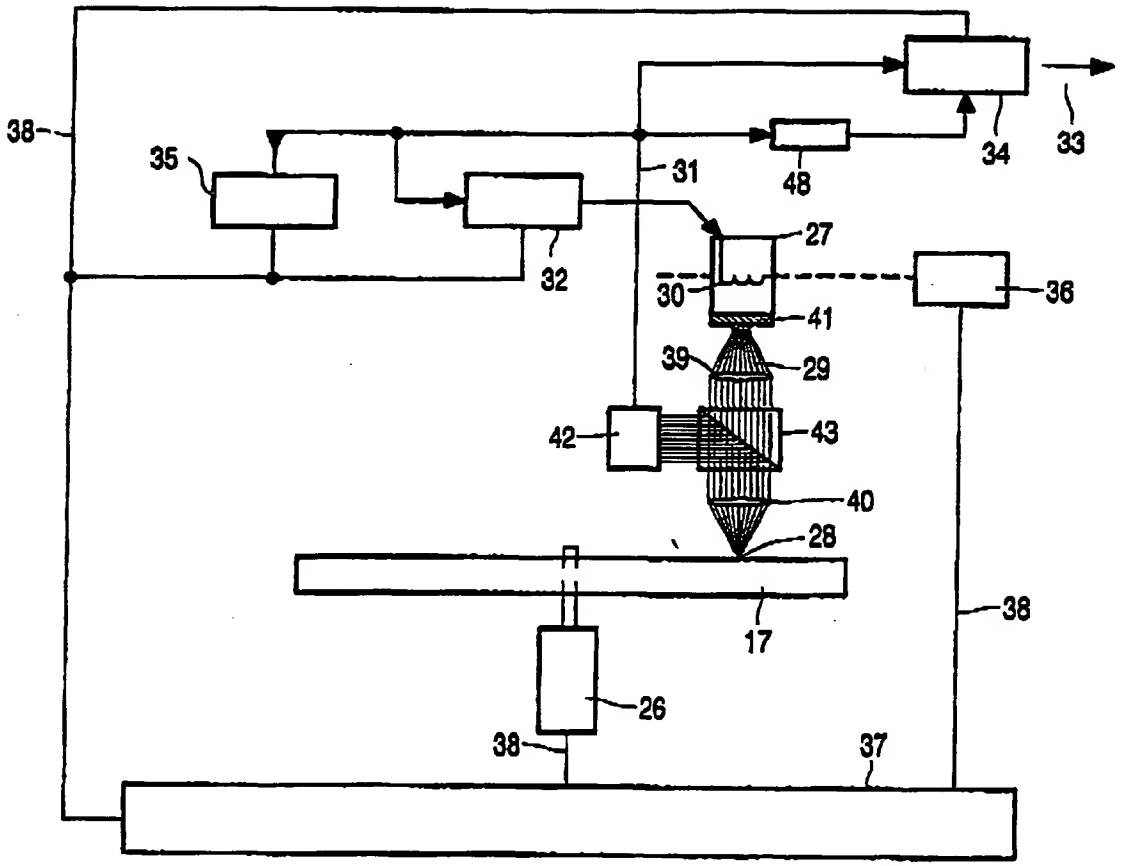

FIG. 1

PH-NL000262

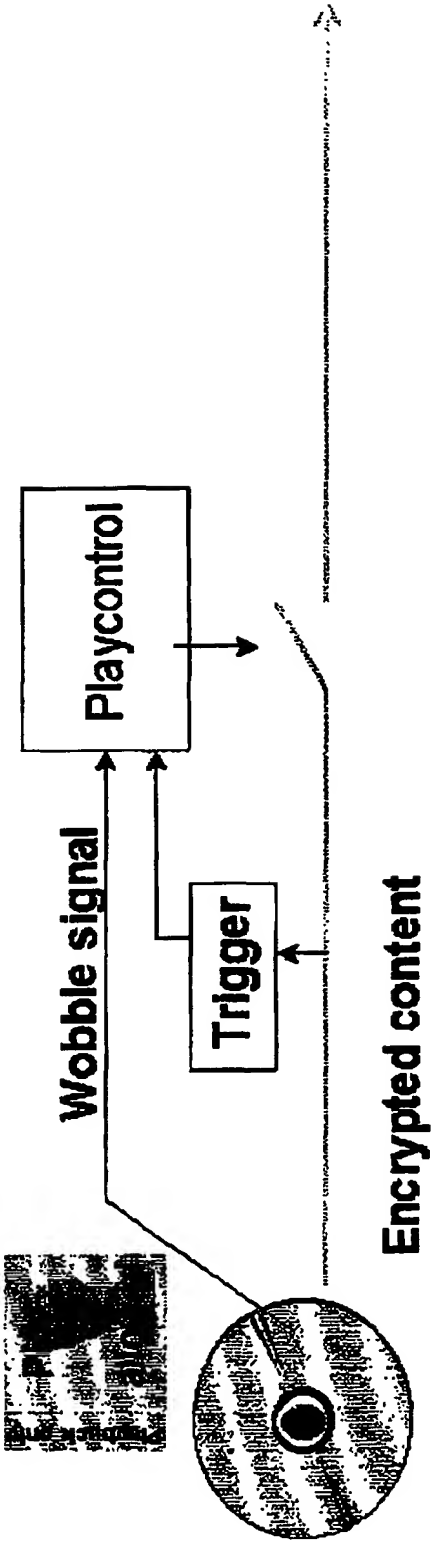

Fig. 2

PH-NL 000262

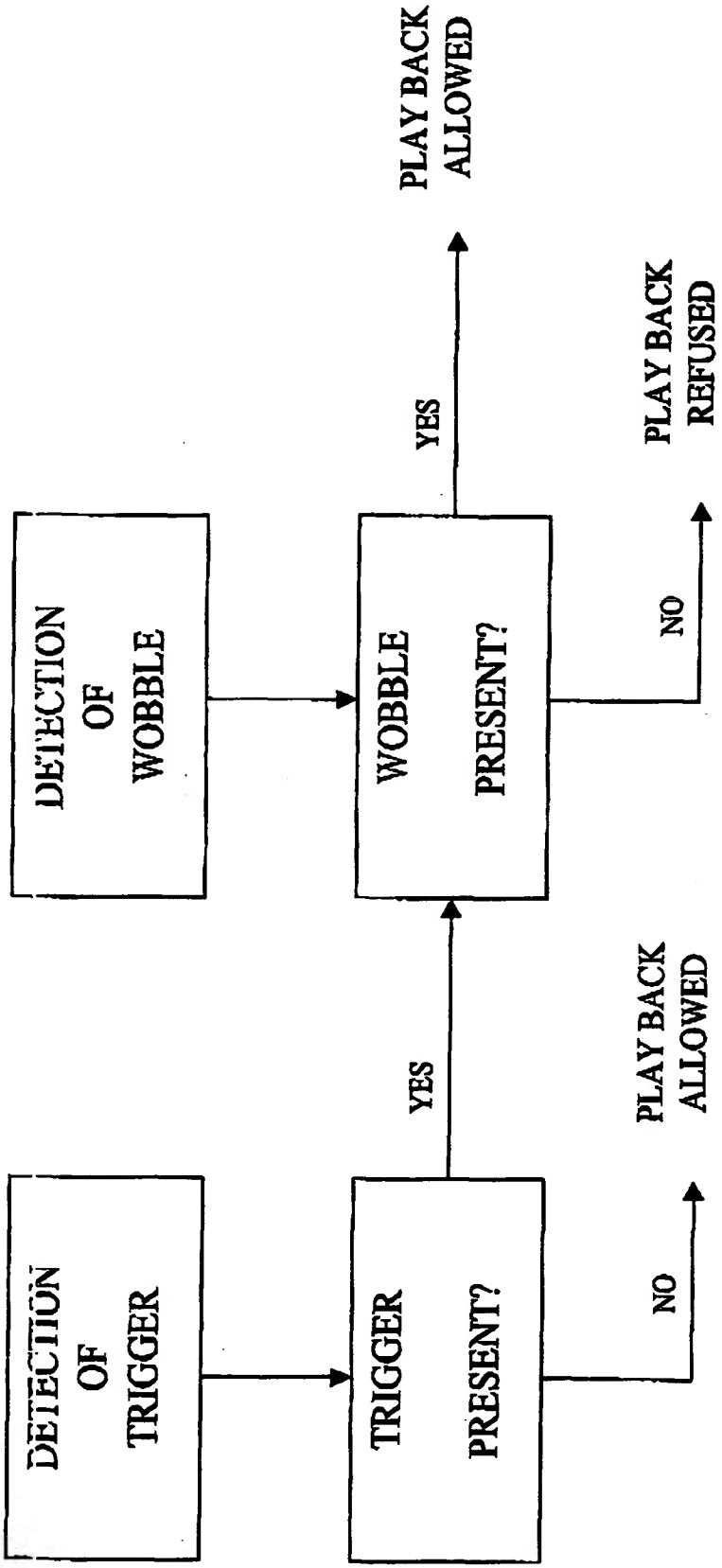

FIG. 3

PHI-NL000262